

RELATÓRIO TP #1 – INTRODUÇÃO À ROBÓTICA

Grupo 8 – Wall-E

Carlos Eduardo
Eugênio
Everton
Julia Rezende
Mônica Emediato

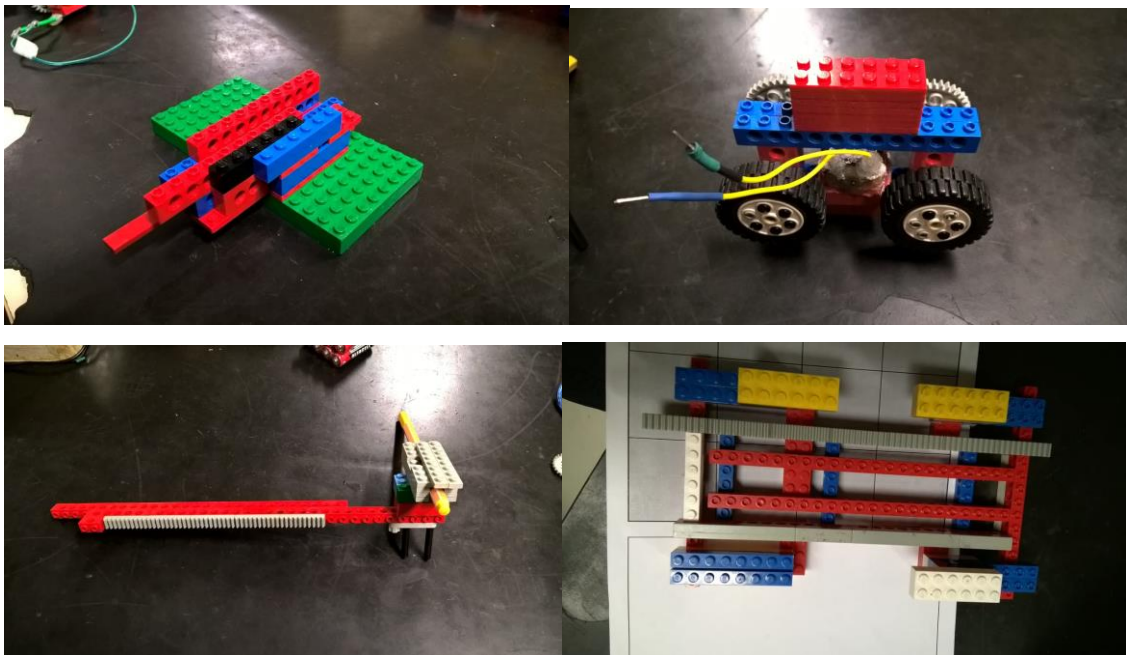
Dividimos a documentação do Trabalho Prático #1 em duas partes: pela do diário de bordo abordando a montagem do robô e complementamos falando sobre a base teórica sendo aplicada.

- Diário de bordo

Nosso grupo se reúne semanalmente desde o anúncio do trabalho prático, pelo menos uma vez por semana.

Na reunião #1, houve a discussão sobre a montagem e a programação do robô. Definimos tarefas para testar os motores e qual TP iríamos fazer, optamos pelo TP2.

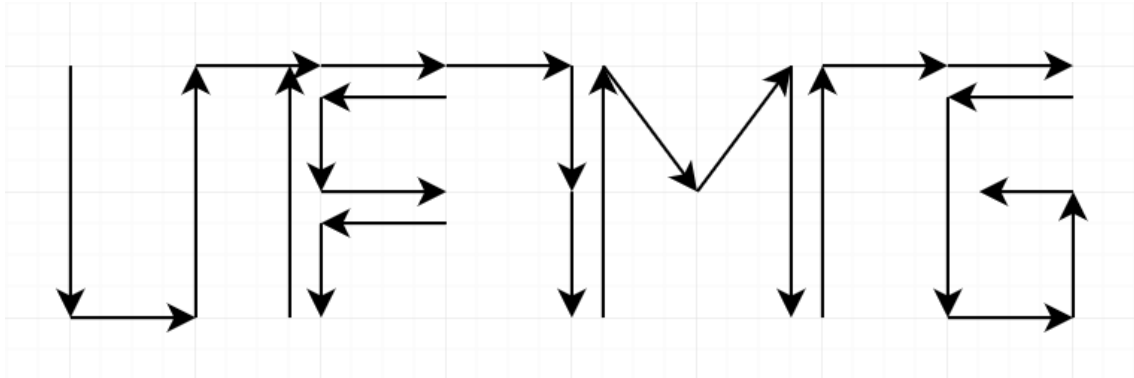
Em nosso segundo encontro, já iniciamos a montagem da parte do lego para que houvesse estabilidade nas peças.



Primeiro protótipo e suas partes

Na reunião #3 o robô estava quase pronto, a base está no limite especificado do TP2. Houve a testagem de ambos os motores para ver se estrutura estava firme. Os avanços serão nessa parte, melhorar o balanceamento de peso.

O trajeto da caneta e o movimento dos braços foi esboçado para que fosse traduzido em forma de programa.



Trajeto da caneta idealizado

Foram feitas algumas modificações na estrutura do robô para otimizar seu funcionamento e dar maior estabilidade. Após esses ajustes começamos a parte da programação, desenvolvemos uma classe representando um movimento que o robô pode fazer, sendo o programa uma sequência de instruções de movimentos, obtidas por nós através de um esboço do desenho e tentativas e erros, uma vez que para realizar o movimento de ida e volta, as potências não eram simétricas, na ida o motor precisava de mais potência que para voltar por conta da divisão de peso. Já conseguimos fazer com que o robô movimente à ponto de escrever as letras U e F.

```

class Movement{
public:
    Adafruit_DCMotor* motor;
    uint16_t force;
    uint8_t mode;
    int delaytime;

    Movement(Adafruit_DCMotor* motor, uint16_t force, uint8_t mode, int
delaytime);
    void execute(void);
};

Movement::Movement(Adafruit_DCMotor* motor, uint16_t force, uint8_t mode, int
delaytime){
    this->motor = motor;
    this->force = force;
    this->mode = mode;
    this->delaytime = delaytime;
};

void Movement::execute(void){
    this->motor->setSpeed(this->force);
    this->motor->run(this->mode);
    delay(this->delaytime);
};
(...)
Adafruit_DCMotor *mY = AFMS.getMotor(2);
Adafruit_DCMotor *mX = AFMS.getMotor(3);
(...)
void printUfmg(){
    (...)
    Movement(mY, 0, RELEASE, 1000).execute();
}

```

Classe Movement e um exemplo de uso

O loop do programa também foi ajustado para executar a sequência apenas uma vez, e depois ficar inativo por tempo indeterminado (até o usuário apertar o botão RST do shield de LED de novo, o que deve reiniciar o programa).

```

void setup() {
  AFMS.begin(); // create with the default frequency 1.6KHz
  lcd.begin(16, 2);
  ledPrint(0,0, "Destroyer v1.0");
  ledPrint(0,1, "ufmg");
  loopLock = false;
}

void loop() {
  if(!loopLock){
    ledPrint(5,1, "(EXEC)");
    printUfmg(); // Toda a sequência de movimentos
    loopLock = true;
    ledPrint(5,1, "(FIM) ");
  }
  delay(1000);
}

```

Funções setup e begin

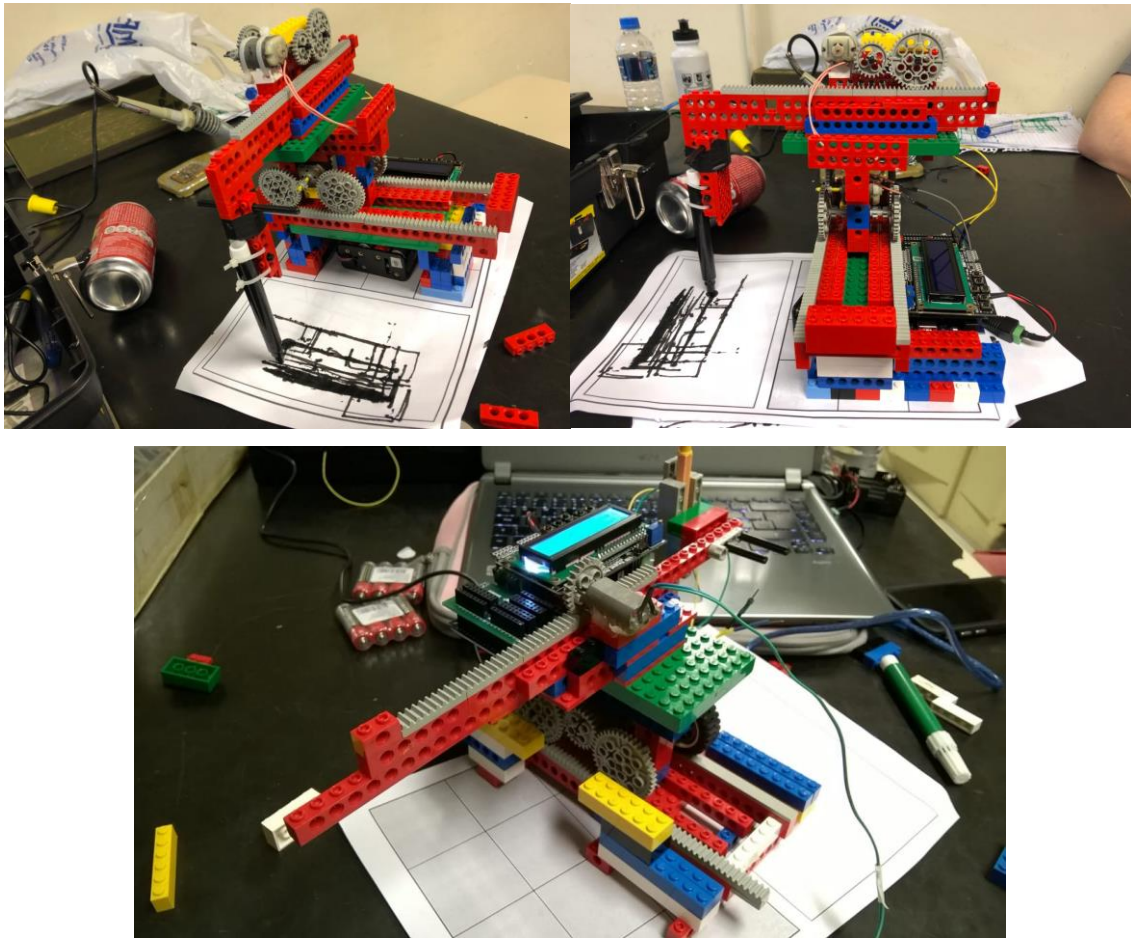
Por fim, realizamos algumas modificações à fim de melhorar o desempenho e a precisão do robô, ele escreve UFMG de modo mais estático pela forma que organizamos sua base no lego e a programação, que possibilitou, por meio de **várias tentativas** que a escrita ficasse simétrica e estável.

Infelizmente alguns contratempos no trabalho aconteceram e tivemos que fazer algumas mudanças as quais não foram possíveis estabilizar o programa de novo há tempo:

O motor do carro inferior queimou na sexta-feira à noite antes da data da entrega, ele teve que ser substituído segunda de manhã e todas as linhas referentes ao seu movimento foram recalculadas rapidamente.

A caneta também esgotou e teve que ser trocada três vezes, também exigindo mudanças no programa (a estabilidade dela muda bastante o comportamento do robô em execução).

O robô escreve UFMG, com variações à cada execução.



Versão final do robô

- Base teórica

Os motores DC têm alta velocidade e baixo torque, por conta disso, utilizamos engrenagens para substituir velocidade por torque. Entretanto, não foram muitas, pois o resultado final foi feito através de poucas transmissões de movimento e deslocamento de potência. Além disso, foi possível programar o robô para que ele executasse com menos potência determinados movimentos sem alterar o resultado final.

Notamos as forças de atrito sendo um principal desafio para a montagem do robô, sendo que o atrito útil nos ajudava, pois ele garante a realização do movimento, e o indesejável, às vezes sendo a causa de movimentos inesperados e indesejáveis.

Além disso, sabe-se que atrito estático é a força de atrito aplicada sobre um corpo enquanto ele não se moveu e a f força de atrito estático máxima é uma força limite de atrito na qual o objeto está prestes a sair do repouso. Por outro

lado, a força de atrito cinético é a força de atrito aplicada sobre um corpo quando ele está em movimento, deslizando sobre uma superfície. Nesse trabalho prático realmente vimos que o coeficiente de atrito cinético é menor que o estático, pois para retirar o robô do repouso, a gente impulsionava-o mais.

Sobre a parte mecânica, decidimos que o Arduino e as pilhas ficariam em baixo da nossa base, o espaço foi montado para que os fios ficassem organizados e os motores tivessem um espaço maior para fazer os movimentos horizontais e verticais.

- Conclusão

O robô ficou difícil de controlar devido às escolhas feitas quanto a configuração das transmissões do motor de baixo (o carro onde fica apoiado o braço). Com a dificuldade no controle, foi difícil fazer um programa que conseguisse mapear o circuito desejado para a realidade, pois o mesmo código traduzia-se de várias maneiras diferentes nas tentativas e testes feitos com o robô no ambiente real.

Para esse caso era desejável eixos mais lentos, porém com tração e ritmo constante, invés de rápidos com arranque forte que se traduziam em diferentes distâncias percorridas dependendo da configuração física do robô (onde ele está no trilho, como está o braço que segura a caneta, como está a situação de equilíbrio da caneta na folha).

O trabalho prático foi nossa primeira experiência montando um robô e agregou bastante tanto para a parte de entrosamento do grupo quanto para nossa prática e conhecimento na construção de robôs simples. Também serviu de introdução ao conceito de controle, nos permitindo experimentar a diferença entre a programação no mundo digital e como essa se manifesta no mundo analógico real, dependendo da mecânica do robô e de seus componentes para ser previsível e executar o que é desejado.